CLAIMS

What is claimed is:

1. A MAP decoding method, comprising the steps of:

performing a first sliding window operation in a first direction on at least a partial block of data, to thereby obtain first derived parameters;

performing a second sliding window operation in a second direction, which is opposite to said first direction, on at least a partial block of said data, to thereby obtain second derived parameters; and

processing said first and second derived parameters, to thereby generate data estimate values;

wherein said sliding window operations are pipelined with each other, to operate in parallel on different respective portions of data.

2. The method of Claim 1, wherein the sliding window operations are each divided into separate stages, and the separate stages operate in parallel on different partial blocks of data.

3. A method for bi-directionally processing a block of data, which does not necessarily have a known state at endpoints thereof, according to at least one sequencing constraint, comprising the steps of:

sequentially processing data elements of the block in a first direction, after first processing prolog elements in said first direction in accordance with said sequencing constraint; and

sequentially processing said data elements in a second direction, after first processing prolog elements in said second direction in accordance with said sequencing constraint.

4. The method of Claim 3, wherein the processing of data elements in the first direction, and the processing of data elements in the second direction are done in parallel.

5. The method of Claim 3, wherein each step of processing data elements is divided into separate stages, and the separate stages operate in parallel on different data elements.

6. A method for parallel MAP processing of a lattice-coded block of data, comprising the steps of:

dividing the data into sliding window blocks, and, for each of multiple ones of said sliding window blocks,

a) sequentially processing the elements of the respective sliding window block in a first direction, after first processing prolog

elements in said first direction in accordance with a sequencing constraint; and

b) sequentially processing the elements of the respective sliding window block in a second direction, after first processing prolog elements in said second direction in accordance with said sequencing constraint;

wherein said steps a) and b) are performed at least partly in parallel with each other.

7. The method of Claim 6, wherein steps a) and/or b) are divided into separate stages, and the separate stages operate in parallel on different sliding window blocks.

8. A method for parallel MAP processing, comprising the steps of:

a) combining probability metrics in at least one adder tree; and

b) performing an maximum-finding operation to combine ones of said metrics which correspond to alternative possibilities;

wherein said steps a) and b) are at least partly performed in a parallelized pipeline relationship with each other.

9. The method of Claim 8, wherein the maximum-finding operation is an exponent-logarithm equation.

10. The method of Claim 8, wherein the maximum-finding operation is an estimation of an exponent-logarithm function.

11. A method for parallel MAP processing, comprising the steps of:

a) combining probability metrics in at least one adder tree;

b) performing a maximum-finding operation to combine ones of said metrics which correspond to alternative possibilities;

c) performing a normalization operation on the results of said step b);

wherein said steps a), b), and c) are at least partly performed in a parallelized pipeline relationship with each other.

12. The method of Claim 11, wherein the maximum-finding operation is an exponent-logarithm equation.

13. The method of Claim 11, wherein the maximum-finding operation is an estimation of an exponent-logarithm equation.

14. A system for MAP processing of a data stream, the data stream being divided into sliding window blocks, comprising:

an alpha generation process;

a beta generation process;

wherein the alpha generation process and the beta generation process are divided into multiple pipelining stages to operate on multiple sliding window blocks using alpha and beta prologs.